Background Estimation with the ABCD Method

Featuring the `TRooFit` Toolkit

W. Buttinger

October 17, 2018

Contents

---

1 Introduction

The *ABCD* method of background estimation is used by many physics analyses at the LHC that search for new physics or even measure rare Standard Model processes. This document is intended to serve as a guide to this method, and attempt to steer the reader towards use of a likelihood-based approach to the method.

A model building toolkit called `TRooFit` (pronounced: "true-fit") is introduced in this guide, and the examples of its use in this document are intended to encourage the reader to construct likelihood models applicable to their particular analysis. Due to the wide variety of scenarios that the ABCD method can be applied to, it is not feasible to provide a one-size-fits-all tool that can be fed with

information and spit out a background prediction. Instead, the `TRooFit` *toolkit* is supposed to allow each individual analysis to apply the techniques described in this guide to their specific case.

In this sense, the first part of this guide (chapter 2) is a review of the usual concept of the ABCD method, and the second part (chapter 3) is a tutorial in statistical model building and fitting, in the context of background estimation.

1.1 A canonical example for the ABCD method

The ABCD method requires that there are two selections that form part of the definition of the signal region, *region A*, which can be inverted in order to define three further regions, *region B, C, and D*. These control regions should ideally be rich in the events produced from background processes that we are trying to estimate with the method.

The selections that are inverted can be one that is applied to a continuous observable (e.g. inverting a requirement on a certain magnitude of missing transverse momentum of the event), or it could be a discrete binary requirement (e.g. a veto on additional leptons in the event). Since a requirement on a continuous observable can be viewed in a discrete pass/fail manner, using selections on continuous observables can be thought of as a more general example and are therefore adopted in this canonical example.

In this example, we suppose there are two continuous observables, v_1 and v_2 that are used to define the signal region and three control regions:

- Region A (signal region): $v_1 > 60$ and $v_2 > 50$,
- Region B: $v_1 < 60$ and $v_2 > 50$,
- Region C: $v_1 > 60$ and $v_2 < 50$,
- Region D: $v_1 < 60$ and $v_2 < 50$,

The cuts were chosen in some manner to preferentially select signal events in the signal region, and have minimal contamination of the other regions with signal events.

We suppose the analysis has blinded the signal region and recorded the data present in the control regions, which is shown in black in figure 1. The figure also shows, in red, a hypothesised shape of the signal that is being targeted by this analysis. The normalization (i.e. the strength) of the signal is often unknown in searches for new physics, so in the traditional arthimetic approaches described in chapter 2 this **signal strength is often assumed to be sufficiently small to render negligible the signal contribution to regions B, C, and D**. We will see in chapter 3 that this assumption is not often necessary for the likelihood based approach.

It could be argued that for the example illustrated here, a more sophisticated selection that cuts along a diagonal in this $v_1 - v_2$ plane would be more appropriate than the horizontal and vertical cuts indicated. However, this guide does not concern itself with issues surrounding cut optimization, which is a whole other area of experimental methods. We will accept the cuts as a given and will address the problem at hand - estimating the background (i.e. non-signal) contribution to the signal region.

The fully binary case (inverting discrete selections) is visualised as the version of this example where only two bins are used per axis (one for pass and one for fail of each selection). Of course, it is possible for one axis to correspond to a continuous observable, and the other to correspond to a binary one. Some of the techniques described in this guide will only be applicable when there is at

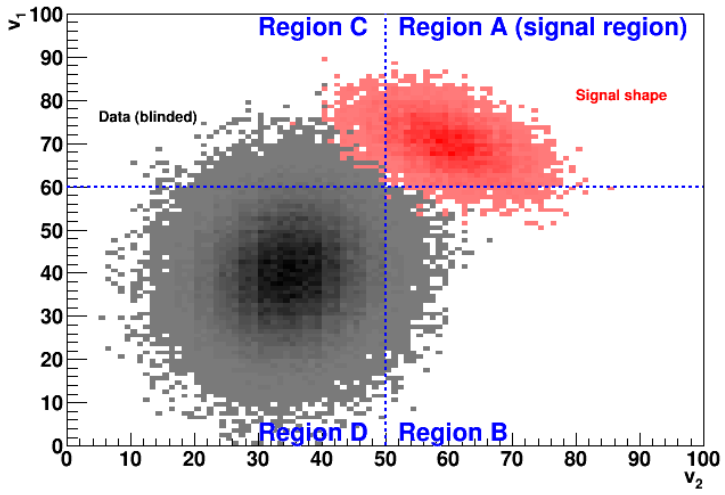


Figure 1: Illustration of a canonical example where the ABCD backgrond estimation method is to be used. The black distribution is the collected data, and the red distribution is the hypothesized shape of the signal that the analysis is targetting.

least one continuous observable available, so in this respect **the use of continuous observables for defining the ABCD regions is preferred**.

The goal of the ABCD method is to produce a prediction for the number of non-signal events in the signal region. Chapter 2 will walk through a traditional arithmetic approach to answering this question, before we attempt in chapter 3 to cast the problem in terms of a statistical likelihood model which we will simply *fit* to the data in order to estimate the background.

2 Traditional arithmetic approach

In this chapter we go through the traditional process of applying the ABCD method to the canonical problem laid out in section 1.1. This will include applying the standard arithmetic ABCD calculation to obtain a prediction, and then attempting to validate this method using a validation region. Typical methods of defining additional systematic uncertainty to the prediction following this validation (in the case of non-closure) as well as attempts to improve the prediction will also be discussed.

2.1 The most basic ABCD prediction

The assumption that underpins the ABCD method is that the following statement is true:

$$\frac{N_C^{\text{bkg}}}{N_D^{\text{bkg}}} = \frac{N_A^{\text{bkg}}}{N_B^{\text{bkg}}} \tag{1}$$

This will be satisfied if the observables defining the ABCD plane are sufficiently uncorrelated for background events. A visual inspection of figure 1 might lead us to believe that this should be approximately true: the data distribution (which we assume is purely background events) looks approximately uncorrelated in the two variables. Therefore proceeding with the ABCD method

seems appropriate. It does not matter for the method that there is an obvious anti-correlation in the signal distribution. **The ABCD method only requires equation 1 to be true for the background distribution that is being estimated**.

We assume that all the events recorded in region B, C, and D, are background events. If we had a reliable theoretical prediction for the signal strength (or for any background that we did not want to be estimated via the ABCD method) then the contribution from this process can be subtracted from the recorded number of events in order to give the number of background events in each region, i.e:

$$N_i^{\text{bkg}} = N_i - N_i^{\text{sig}}, \text{for } i = B, C, D \tag{2}$$

However, in this example we will assume **we do not have a reliable prediction for N_i^{sig} and will therefore assume that they are negligible, i.e. that $N_i^{\text{bkg}} = N_i$.**

The background estimate for the signal region, N_A^{bkg} is obtained from rearranging equation 1:

$$N_A^{\text{bkg}} = \frac{N_C^{\text{bkg}}}{N_D^{\text{bkg}}} N_B^{\text{bkg}} = \frac{N_C}{N_D} N_B. \tag{3}$$

For the canonical example, the following numbers are given:

$$\begin{aligned} N_B &= 1557 \\ N_C &= 2249 \\ N_D &= 96131 \\ \implies N_A^{\text{bkg}} &= 36.4 \pm 6.0 \text{ (stat)} \pm 1.2 \text{ (syst)}, \end{aligned} \tag{4}$$

where the statistical uncertainty is the standard poisson uncertainty on the nominal prediction, and the systematic uncertainty is obtained from normal error propagation of the statistical uncertainties on measurements of the (true but unknown) background rates in regions B, C, and D.

2.2 Validating the basic ABCD method

We can define a validation region if we have at least one non-binary observable defining the ABCD plane (the observable could be discrete, such as *number of jets*, we just require a way to divide the plane into six regions now, instead of four[1].

Figure 2 shows one possible way to define a validation region: regions B and D are subdivided into regions B', A', D', and C'. The ABCD method can then be applied to these four regions, where an attempt is made to estimate the background yield in region A' (the validation region), using the other three regions.

For the canonical example, this leads to:

[1] It is possible to extend a binary observable by incorporating another (binary or otherwise) selection requirement into the axes variable: i.e. go from bins representing (A,!A) to (A&&B, A&&!B, B&&!A, !A&&!B) or even just (A, !A&&B, !A&&!B)

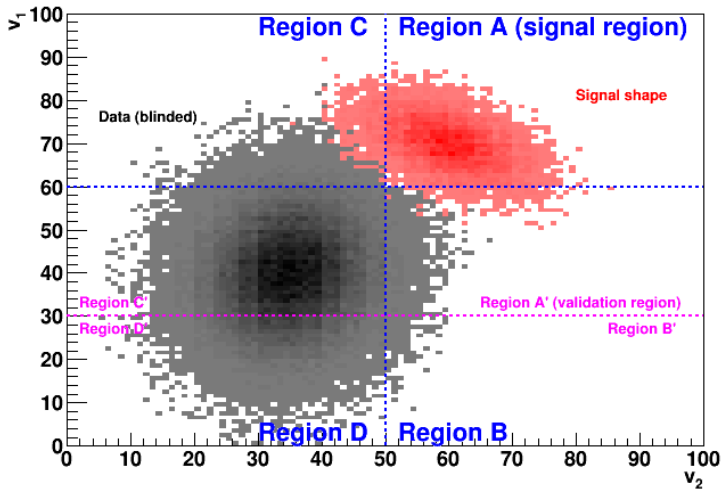


Figure 2: Region B and D of the ABCD plane can be cut into two, to define new regions B',A', D', and C'. Region B' can be used as a validation region in which to test the ABCD method.

$$\begin{aligned} N_{B'} &= 192 \\ N_{C'} &= 80379 \\ N_{D'} &= 15752 \\ \implies N_{A'}^{\text{bkg}} &= 979 \pm 31 \text{ (stat)} \pm 71 \text{ (syst)}, \end{aligned} \tag{5}$$

However, in the canonical example the number of observed events in region A' is 1365 events. This suggests there might be a problem with the underlying assumption of the ABCD method (equation 1) in this case. The following sections will describe a ways to proceed at this point.

2.3 Adding a non-closure uncertainty

The crudest approach to this problem is to assign a non-closure uncertainty, given by the relative difference between the prediction and the observed events in the validation region. In this case the relative uncertainty would be 39%. When taking this approach, a statement should be made regarding the change in the signal sensitivity as a result of the additional uncertainty on the background.

You also should then attempt to validate your non-closure. This involves defining a new validation region (and accompanying control region) in which to check that your ABCD prediction with additional non-closure uncertainty will cover the observed number of events in this new validation region. This means we are now up to requiring 8 regions:

1. Signal region (region A).
2. Accompanying control region (region B).
3. Primary validation region, where non-closure is discovered and an uncertainty is measured (region A').
4. Accompanying control region of the primary validation region (region C').

5. Region giving the numerator of the transfer factor for the validation (region B').
6. Region giving the denominator of the transfer factor for the validation (region D').
7. Secondary validation region, to test the non-closure uncertainty
8. Accompanying control region of the secondary validation region

In this guide we have not made all these regions orthogonal: we defined regions 2-4 with subregions of the regions we ultimately were using as the the control region and transfer factor denominator for the main background estimation of region A (the numerator coming from region C). **It should be discussed whether it is appropriate to use non-orthogonal regions in this process**.

Finally, whatever secondary validation region you choose, ideally the prediction should have a similar order-of-magnitude background prediction as your final signal region. So in this canonical example, we would choose a secondary validation region that has a prediction of O(10) events (given our signal region background prediction is approximately 36 events).

2.4 Improving the prediction

This improvement is only really possible if at least one of the ABCD plane axes is defined by a continuous observable. In this example we will utilise that v_2 is continuous, since we used v_1 to define our validation region.

Figure 3 shows three alternative ways to define a subregion of region C and D (the transfer factor numerator and denominator regions): the black-shaded region of $0 < v_2 < x$, the blue-shaded region of $x < v_2 < 50$, and the red-shaded region of $x - 5 < v_2 < x + 5$. The transfer factor can be measured as a function of x, and the result of this is shown in figure 4. A clear trend in the transfer factor is seen, and it is reasonable to believe that the transfer factor defined closest to the signal region (i.e. to the right of the plot) is closest to the "correct" transfer factor that should be applied to the control region (region B or B') in order to obtain the prediction for the signal or validation region (region A or A').

Using the transfer factors, N_{C}/N_{D} and $N_{C'}/N_{D'}$, measured nearest the signal and validation regions gives:

$$\begin{aligned} \frac{N_{C'}}{N_{D'}} &= 6.64 \pm 0.13 \\ \implies N_{A'}^{\text{bkg}} &= N_{B'}\frac{N_{C'}}{N_{D'}} = 1275 \pm 35 \text{ (stat)} \pm 95 \text{ (syst)} \text{ (observed 1365)} \end{aligned} \tag{6}$$

$$\begin{aligned} \frac{N_{C}}{N_{D}} &= 0.032 \pm 0.001 \\ \implies N_{A}^{\text{bkg}} &= N_{B}\frac{N_{C}}{N_{D}} = 49.5 \pm 7.0 \text{ (stat)} \pm 2.3 \text{ (syst)} \end{aligned} \tag{7}$$

$$\tag{8}$$

A new non-closure systematic could also be assigned at this point, however the uncertainty on the estimate in the validation region covers the observation, so an additional systematic may be deemed unnecessary in this case.

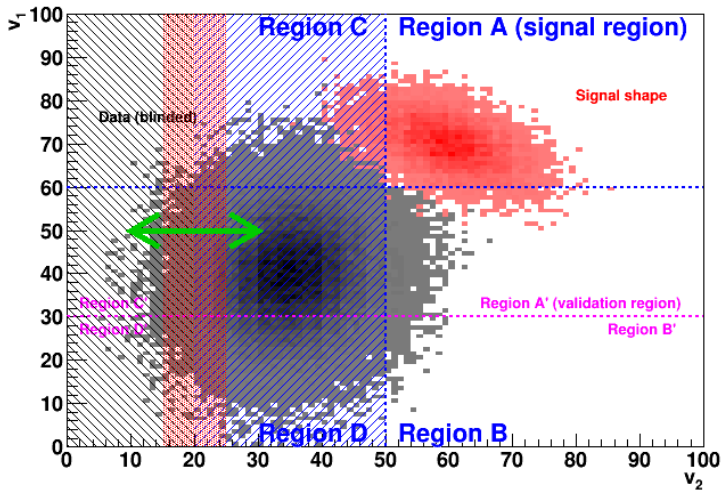
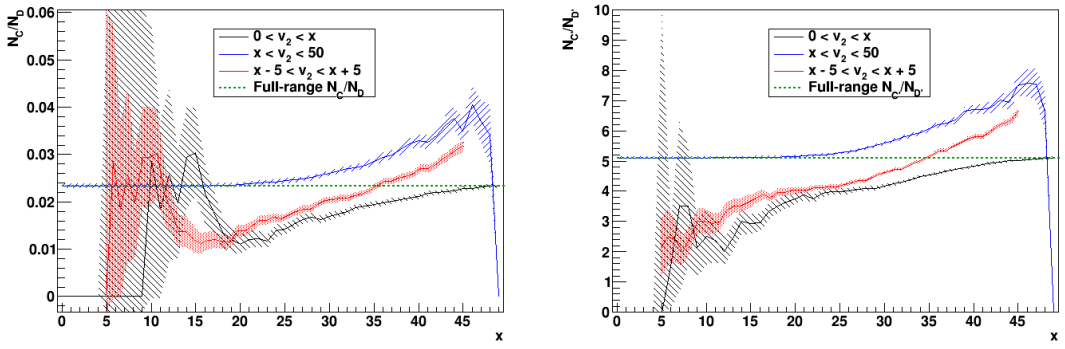


Figure 3: The ratio N_C/N_D (and $N_{C'}/N_{D'}$) can be measured in subregions of region C and D (C' and D') when the v_2 variable is sufficiently *continuous*. Making this measurement may show evidence of a trend in the data that is inconsistent with the base assumption that $N_C/N_D = N_B/N_A$ for background events. The shaded black region corresponds to $0 < v_2 < x$, shaded blue is $x < v_2 < 50$ and shaded red is $x - 5 < v_2 < x + 5$, where in all cases $x = 20$. The ratios can be measured as a function of x, by varying the subregions as indicated by the green arrows.



(a) Regions C and D

(b) Regions C' and D'

Figure 4: The ratios N_C/N_D and $N_{C'}/N_{D'}$ measured as a function of x for three different definitions of sub-regions of C, D, D', and C'. See figure 3 for further explanation of x. The dashed green lines indicate the ratio obtained from the full regions.

2.5 Summary of arithmetic approach and difficulties

3 Likelihood based approach

A likelihood-based approach to the ABCD method is really just fitting a statistical model that is constructed with an underlying assumption about the relationship of the background distribution between different regions.

The assumption of the basic ABCD method (equation 1) can be expressed as:

$$\tilde{N}_A = \tilde{m}\tilde{N}_B, \tilde{N}_C = \tilde{m}\tilde{N}_D, \tag{9}$$

and the likelihood for observing the data $\mathbf{data} = \{N_A, N_B, N_C, N_D\}$ is given by:

$$\begin{aligned} L(\mathbf{data}|\tilde{N}_B, \tilde{N}_D, \tilde{m}) &= \text{Pois}(N_A|\tilde{m}\tilde{N}_B)\text{Pois}(N_B|\tilde{N}_B)\text{Pois}(N_C|\tilde{m}\tilde{N}_D)\text{Pois}(N_D|\tilde{N}_D) \\ &= \text{Pois}(N_A + N_B + N_C + N_D|\tilde{N}_{\text{tot}}) \prod_{N_A} \frac{\tilde{m}\tilde{N}_B}{\tilde{N}_{\text{tot}}} \prod_{N_B} \frac{\tilde{N}_B}{\tilde{N}_{\text{tot}}} \\ &\quad \prod_{N_C} \frac{\tilde{m}\tilde{N}_D}{\tilde{N}_{\text{tot}}} \prod_{N_D} \frac{\tilde{N}_D}{\tilde{N}_{\text{tot}}}, \end{aligned} \tag{10}$$

where

$$\tilde{N}_{\text{tot}} = \tilde{m}\tilde{N}_B + \tilde{N}_B + \tilde{m}\tilde{N}_D + \tilde{N}_D. \tag{11}$$

The notation used is that free parameters in the fit have a $\sim$ above them. The form of equation 10 is to emphasise that the model should be thought of as a *four-bin* model, where the observable is the bin that each event falls in to. Therefore the probability of observing the data is the product of the probabilities for each event to have fallen into the bin that it did, multiplied by an overall Poisson probability of observing the total number of events that were observed. The splitting of the likelihood into a single overall Poisson term and a series of probabilities of each event is a feature of the `RooFit` model building toolkit. In `RooFit` likelihoods are built from normalized PDFs for the event-level observables (the *region* is the observable in this case), with a Poisson term added when the PDF is an *extended* pdf.

3.1 Counting degrees of freedom

It is important to verify that the number of free parameters is not greater than the number of observations, otherwise the fit would be underconstrained. In this case, there are four observations (N_A, N_B, N_C, N_D) and three free parameters $(\tilde{m}, \tilde{N}_B, \tilde{N}_D)$ so if performing a fit to all four regions we actually have room in our model for one more free parameter. Later on when incorporating signal into the model, we will introduce signal strength as the additional free parameter. However, when performing a fit on blinded data, N_A is not available and is removed from the model:

$$L(\textbf{blinded data}|\tilde{N}_B, \tilde{N}_D, \tilde{m}) = \text{Pois}(N_B + N_C + N_D|\tilde{N}_{\text{blind tot}}) \prod_{N_B} \frac{\tilde{N}_B}{\tilde{N}_{\text{blind tot}}} \prod_{N_C} \frac{\tilde{m}\tilde{N}_D}{\tilde{N}_{\text{blind tot}}} \prod_{N_D} \frac{\tilde{N}_D}{\tilde{N}_{\text{blind tot}}},, \quad (12)$$

with

$$\tilde{N}_{\text{blind tot}} = \tilde{N}_B + \tilde{m}\tilde{N}_D + \tilde{N}_D. \quad (13)$$

3.2 Constructing the basic likelihood model

We now show how to construct this model using the `TRooFit` extension to `RooFit`. The manner in which the model is constructed will allow it to be made more sophisticated in several ways:

- Binning within the individual regions.
- More general relationships between the regions (compared to the simplest relationship defined by equation 9).
- Adding signal shape information to the simultaneous fit.

The model will be constructed with four `TRooH1D`, which should be thought of as a version of a `ROOT TH1D` histogram that have extra features so that it can function as a PDF in the `RooFit` toolkit. We start by constructing these four `TRooH1D`.

```
int nBins = 1; //number of bins per region. Start with 1 bin
double cLeft=0; //left edge of region C
double aLeft=50; //left edge of region A
double aRight=100; //right edge of region A
double cTop=100; //top edge of region C
double cBottom=60; //bottom edge of region C
double dBottom=0; //bottom edge of region D

const char* regionLabels[4] = {"C","A","D","B"};
//odd ordering is so that when we draw regions
//we will get:
// C | A
// D | B

RooRealVar x("v2","v2",cLeft,aRight); //a RooFit continuous variable with range
                                      //cLeft to aRight

TRooH1D* b[4]; //will point to the four TRooH1D we will create

for(int i=0;i<4;i++) {
  b[i] = new TRooH1D(Form("b_%s",regionLabels[i]),
                     Form("Region %s bkg",regionLabels[i]),
                     x,nBins,(i==0||i==2)?cLeft:aLeft,(i==0||i==2)?aLeft:aRight);
  b[i]->SetFillColor(kCyan);
  b[i]->setFloor(true); //prevents 'value' of the histogram being less than 0
}
```

We want to allow the values of the region B and region D `TRooH1D`s to be free parameters in the model. For this we can introduce an additional `RooRealVar` for each bin, and attach them to each bin of the `TRooH1D` using the `addShapeFactor` method. A `shapeFactor` is a scale factor applied to the value of a single bin of a `TRooH1D`. This means that there needs to be a non-zero bin content in order for the scale factor to have an effect. Since we expect the post-fit bin values to be very close to the data measurement in that bin, we can choose to set the bin content equal to the data

measurement (or 2, if the measurement is fewer than 2 events), and then giving the scale factor a range of 0-5 should be adequate to cover the fit solution.

```
for(int j=2;j<=3;j++) {  //j = region index ... region 2 and 3 = D and B
  for(int i=1;i<=nBins;i++) {
        RooRealVar* sf = new RooRealVar(Form("sf_%s_bin%d",regionLabels[j],i),
                                        Form("sf_%s_bin%d",regionLabels[j],i),1,0,5);
    b[j]->SetBinContent(i,hdata[j]->GetBinContent(i));
    if(b[j]->GetBinContent(i)<2) b[j]->SetBinContent(i,2);
    b[j]->addShapeFactor(i,*sf);
  }
}
```

In the above code, `hdata[i]` is a histogram containing the observed data for the i^{th} region, with the same binning as the `TRooH1D`. So far the inner loop will only be over a single bin, but setting up the code in this way will the model to be generalised to multiple bins per region.

The model then requires that value the region A and C PDFs should correspond to the region B and D PDFs multiplied by an additional scale factor, $\tilde{m}$. This can be accomplished with the following code:

```
b[0]->Fill(*b[2]); //makes bin content of region C equal to region D
b[1]->Fill(*b[3]);

//initialize m parameter to ratio of integrals of data histograms
//in regions C and D
//We choose range of m to be between 0 and 5 ...
RooRealVar m("m","#tilde{m}",hdata[0]->Integral()/hdata[2]->Integral(),0,5);

b[0]->addNormFactor(m);
b[1]->addNormFactor(m);
```

A `normFactor` is a scale factor applied to all bins, as opposed to a `shapeFactor` which applies to only a single bin.

At this stage we can inspect what the four regions look like by drawing them, along with the data (except in the signal region (region A, which has index=1)):

```
TCanvas c("Prefit","Prefit distributions",800,600);
c.Divide(2,2);
for(int i=0;i<4;i++) {
  c.cd(i+1);
  b[i]->Draw();
  if(i!=1) hdata[i]->Draw("same");
  if(i==1) {
    //print the TRooH1D integral (i.e. prediction) in signal region
    TText t;
    double bkgErr;
    double bkgPredicition = b[i]->IntegralAndError(bkgErr);
    t.DrawTextNDC(0.4,0.8,Form("Predicted = %g +/- %g",bkgPrediction,bkgErr));
  }
}
```

The result of this code, for the canonical example, is given in figure 5. The histograms in region B, C, and D all line up with the data due to the choice of initial values for the free parameters. The prediction for region A is consistent with the number calculted in equation 4. However, there is no systematic error on the integral because the fit has not been performed yet.

This code allows us to easily change the number of bins in the distributions, for example figure **??** shows the prefit distributions when there are 10 bins per region (`nBins=10`). The small change in the prediction in region A is due to the setting on histogram bin contents in region B and D to never be smaller than 2.

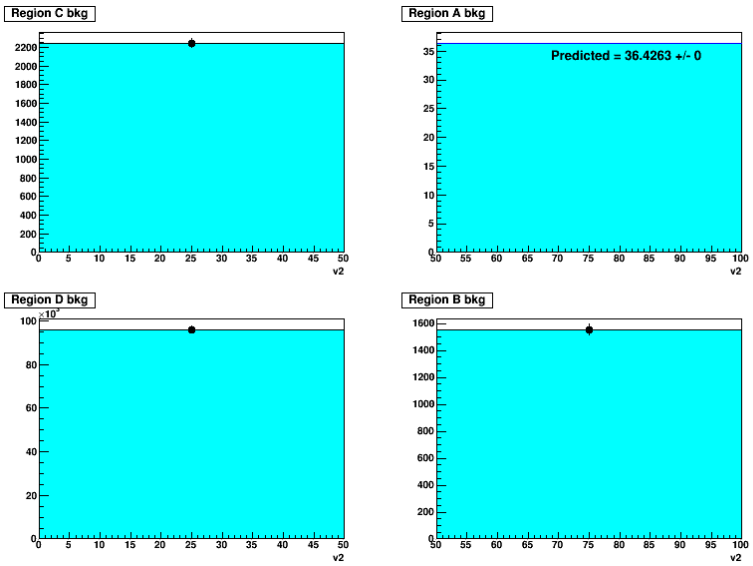


Figure 5: Prefit state of the model when using `nBins=1`, with the data overlaid in regions B, C, and D. The prediction (integral of the blue histogram) in region A is also shown. There is no error because the fit has not yet been performed.

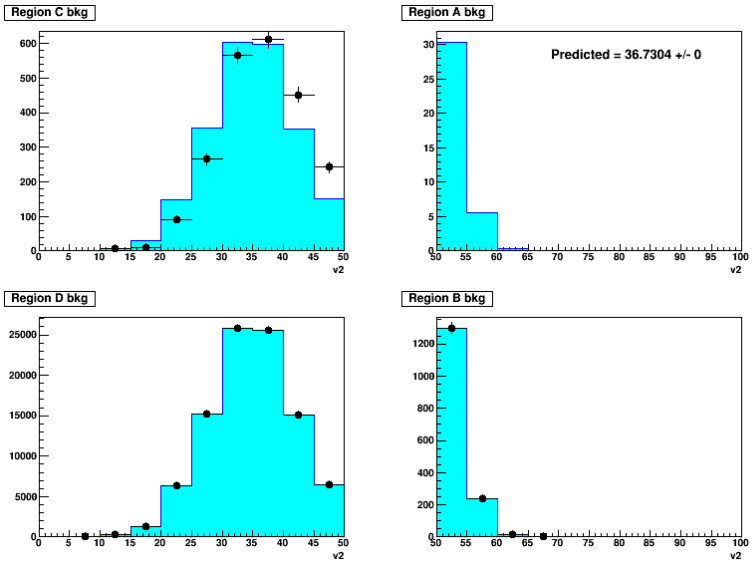


Figure 6: Prefit state of the model when using `nBins=10`, with the data overlaid in regions B, C, and D. The prediction (integral of the blue histogram) in region A is also shown. There is no error because the fit has not yet been performed. The slight difference in the prefit prediction compared to the `nBins=1` case is due to the overriding of bin contents in regions B and D, as explained in the text. The disagreement between the data and model in region C is suggesting that the standard ABCD model is not appropriate in this case.

Finally, we need to bring all four `TRooH1D` together to build a model that is a function of the event-level observable, the *region*. The region observable is represented with a `RooCategory` object:

```
//define a RooCategory to represent which region the event is in
RooCategory cat("region","region");
for(int i=0;i<4;i++) cat.defineType(regionLabels[i]);

//construct the full model
RooSimultaneous model("model","model",cat);
for(int i=0;i<4;i++) {
  model.addPdf( *b[i], regionLabels[i] ); //associates PDF b[i] with ith region
}
```

3.3 Fitting the model to the data

In order to use `RooFit` to fit the model to the data, the data must be placed in a `RooDataSet`. In the following code, the data comes from a `TTree` in a file called `abcdInputTrees.root`.

```
TFile dataFile("abcdInputTrees.root");
TTree* dataTree = (TTree*)dataFile.Get("data");
double v1,v2;
dataTree->SetBranchAddress("v1",&v1);
dataTree->SetBranchAddress("v2",&v2);

RooDataSet data("data","data",RooArgSet(x,cat));
for(int i=0;i<dataTree->GetEntries();i++) {
  dataTree->GetEntry(i);

  if(v1>cTop) continue;
  if(v2<cLeft) continue;
  if(v2>aRight) continue;
  if(v1<dBottom) continue;


  if(v1<cBottom&&v2<aLeft) cat.setLabel("D");
  else if(v1>=cBottom&&v2<aLeft) cat.setLabel("C");
  else if(v1<cBottom&&v2>=aLeft) cat.setLabel("B");
  else cat.setLabel("A");
  x=v2;
  data.add(RooArgSet(x,cat)); //how to add an event to a RooDataSet
}
```

We are just about ready to fit the model to the data now. This would normally be performed by simply doing:

```
RooFitResult* fitResult = model.fitTo(data,RooFit::Save());
```

where the `fitResult` object will contain the post-fit values of the free parameters, along with the uncertainties and correlation information for those parameters. However, this is only appropriate if we are fitting with the unblinded data. If the data is still blinded, a fit should only be performed with regions B, C, and D. To achieve this, the data is *reduced* to remove the events in the signal region, and the signal region of the model (`b[1]`) has its `BlindRange` set so that it effectively is masked out of the model:

```
x.setRange("myRange",aLeft,aRight); //create a RooFit blind range
b[1]->setBlindRange("myRange"); //blind the signal region in this range
RooAbsData* blindedData = data.reduce("region!=1"); //removes the region A data
RooFitResult* fitResult = model.fitTo(*blindedData,RooFit::Save());
b[1]->setBlindRange(""); //remove the blinding range for the post-fit plotting
delete blindedData;
```

The result of the fit is shown in figure 7, where when drawing the `TRooH1D` objects the `"e3005"` option has been used; this option draws a shaded error band (with `FillStyle=3005`). In fact,

to ensure that all the correlations between the free parameters are properly accounted for when calculating the error band, the `fitResult` object should also be passed to the `Draw` method. If this object is not passed, then the errors come directly from the free parameters themselves (`RooFit` copies the post-fit errors onto the parameters), and any correlations between the errors are neglected.

```
b[i]->Draw("e3005"); //draws TRooH1D with a shaded error band
b[i]->Draw("e3005",fitResult); //draws error band, taking correlations into account
```

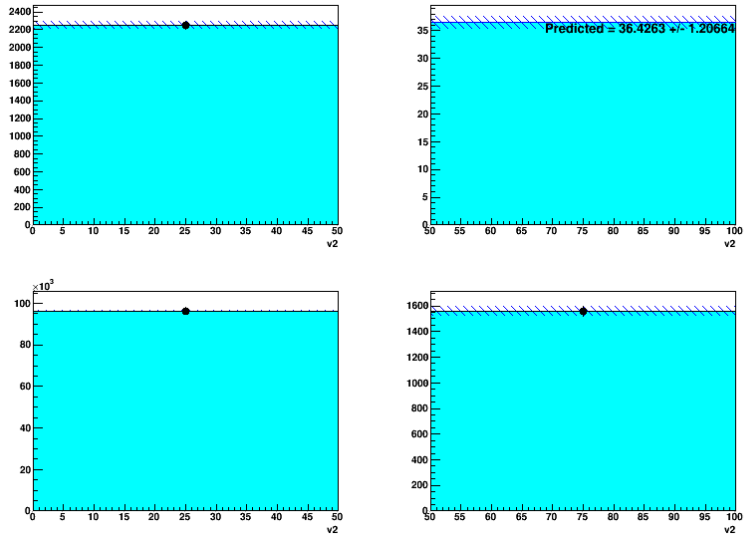


Figure 7: Post-fit distributions, in the `nBins=1` case.

Figure 8 with the additional bins shows that the model is clearly not adequate at describing the data.

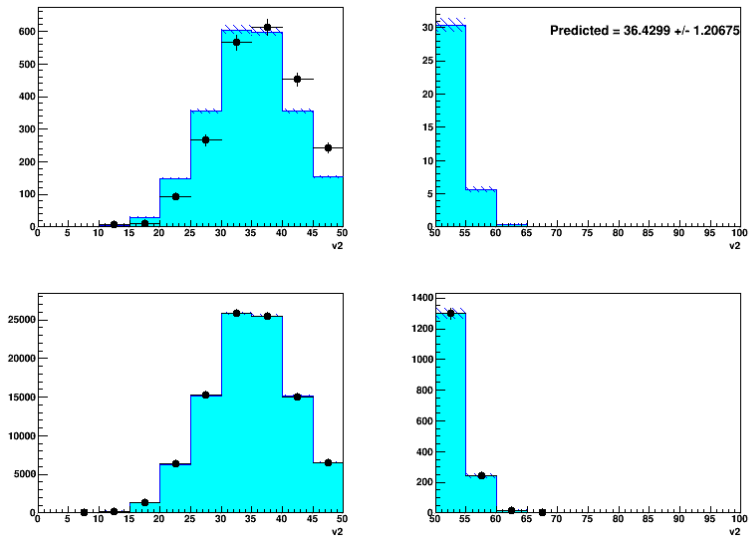


Figure 8: Post-fit distributions, in the `nBins=10` case.

3.4 Validation and non-closure uncertainties

Single bin validation .... (by defining validation regions and fitting in those regions) ... adding a relative uncertainty to region A ... (just adding a constrained normFactor, with uncert given by the validation region relative fit difference)

3.5 Using a linear model

We can attempt to improve the model by replacing the simple ABCD relationship of equation 10 with a relationship that is a function of one of the ABCD plane observables:

$$\tilde{N}_A(x) = \tilde{N}_B(x)(\tilde{m}_1 x + \tilde{m}_2), \tilde{N}_C(x) = \tilde{N}_D(x)(\tilde{m}_1 x + \tilde{m}_2), \tag{14}$$

This naturally only works for continuous observables. Additionally, in order to use this model we cannot have a single bin per region (if fitting in the blinded case) because we now have an additional free parameter in the fit. We will focus on the 10-bin case for now, but any number of bins greater than 1 would in theory be sufficient to provide enough data points to constrain the model.

The linear model can be constructed using standard `RooFit` classes:

```
RooRealVar m1("m1","#tilde{m}_{1}",0.,-5,5); //initial guess is 0 gradient
RooRealVar m2("m2","#tilde{m}_{2})",hdata[0]->Integral()/hdata[2]->Integral(),-5,5);
RooFormulaVar transferFunc("transferFunc",
                           "Transfer Factor as function of v2",
                           "(m1*v2 + m2)",RooArgList(m1,m2,x));
```

In the model construction, instead of adding `m` as a `normFactor` to `b[0]` and `b[1]`, one should use `transferFunc`. This would be sufficient to construct the model.

However, it would also be useful to be able to visualize the `transferFunc` function, as well as ensure that its value can never go negative (since a negative transfer factor would be unphysical). There are a number of ways to achieve both these things, but one way is to use another `TRooFit` class called the `TRooHF1D`. This is similar to a `TRooH1D` except that it represents a function rather than a PDF, i.e. its value is not a density, and it is not a normalized distribution. In this example, a separate `TRooHF1D` has been used for each pair of regions, so that they are easier to plot individually.

```
TRooHF1D tDC("tDC","Transfer from region D to C, as function of v2",
             x,nBins,cLeft,aLeft);
tDC.Fill(transferFunc); //uses the function as its value
tDC.setFloor(true); //prevents going negative, which would be unphysical


TRooHF1D tBA("tBA","Transfer from region B to A, as function of v2",
             x,nBins,aLeft,aRight);
tBA.Fill(transferFunc); //uses the function as its value
tBA.setFloor(true); //prevents going negative, which would be unphysical

//later on ...
b[0]->addNormFactor(tDC); //instead of b[0]->addNormFactor(m);
b[1]->addNormFactor(tAB); //instead of b[1]->addNormFactor(m);
```

When the fit is rerun with this model, a very different prediction is obtained, as shown in figure 9.

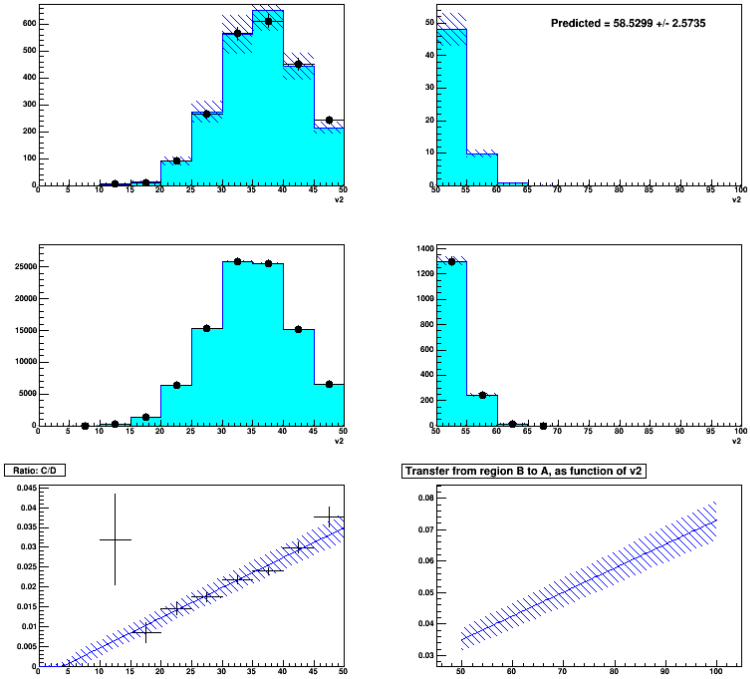


Figure 9: Post-fit distributions, in the `nBins=10` case, with a linear model used for the transfer factor from region D (B) to C (A). The bottom two graphs so the post-fit values of the transfer factor (compared to the ratio of the data in regions C and D on the left hand side).

4 ABCD method checklist

Analyses using the ABCD method are required to define a minimum of six regions: the four main regions, and two additional regions with which to perform a validation of the method. More may be required, depending on the success of the validation.

1. Define signal region (region A), accompanying control region (region B), and transfer factor measurement regions (region C and D).
2. Define your transfer factor model for the background. The two options explored in this guide were:
   - Standard ABCD assumption i.e. constant transfer factor (m):
   $$N_A = mN_B, N_C = mN_D \tag{15}$$
   - Linear transfer factor model (valid for continuous variable x):
   $$N_A(x) = N_B(x)(m_1 x + m_2), N_C(x) = N_D(x)(m_1 x + m_2) \tag{16}$$

   Another option, not explored in this guide, but known to have been used by some analyses, is:
   - Adjusted flat model:
   $$N_A = \rho m N_B, N_C = mN_D \tag{17}$$
   where ρ is a nuisance parameter constrained by an observable corresponding to the estimate of ρ from, e.g., MC simulation (estimate ρ from $(N_A^{\mathrm{MC}}/N_B^{\mathrm{MC}})/(N_C^{\mathrm{MC}}/N_D^{\mathrm{MC}})$).

3. If including signal (only possible when enough measurements are defined) with an unknown signal strength, run the fit with signal strength floating, ensuring that the range of the strength parameter is sufficiently large to cover the assumption that all the data in the control regions is due to signal. Repeat the fit with signal strength fixed to 0. Take the difference between the signal region predictions as an uncertainty on the background predictions due to uncertain signal strength.

4. Define a (primary) validation region (region A'), accompanying control region (region B'), and transfer factor control regions (regions C' and D'). Figure 10 shows some examples of possible control regions that can be defined when the ABCD-plane variables are continuous. It is also possible to define validation regions by inverting a binary selection.

5. If non-closure (beyond statistical fluctuations) is observed in the validation region, either improve the model, or assign a non-closure uncertainty equal to the non-closure relative to the prediction.

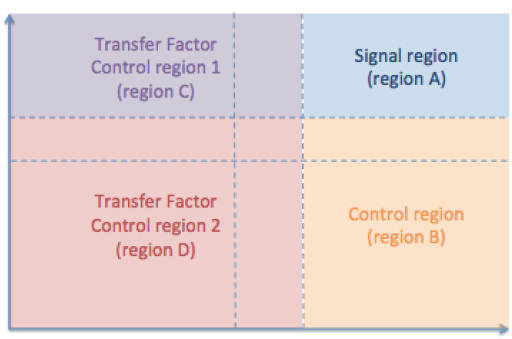
6. If a non-closure uncertainty is added, define a secondary validation region and accompanying control regions. These regions should be orthogonal to the primary validation and accompanying control regions, and the prediction in the secondary validation region should be the same order of magnitude as the prediction in the signal region.

7. Confirm that the prediction with non-closure uncertainty adequately covers the observation in the secondary validation region.

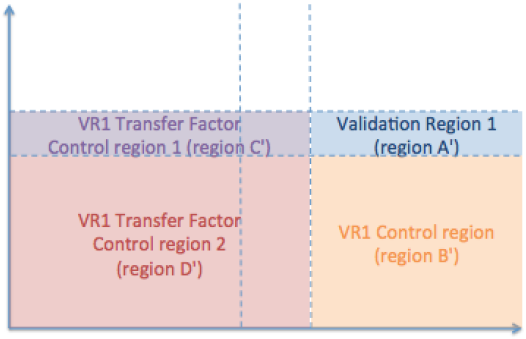
8. Additional validation regions may be defined in order to improve confidence in the estimate.

9. Where continuous variables have been used for both axes of the ABCD plane, swapping over the variables used to define the regions is a good way to cross-check the prediction.

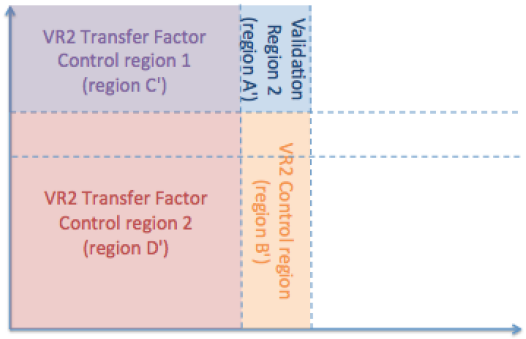
Figure 10 shows suggestions of how to define possible validation regions. An alternative way to define validation regions is to invert an analysis cut to define a validation ABCD plane.



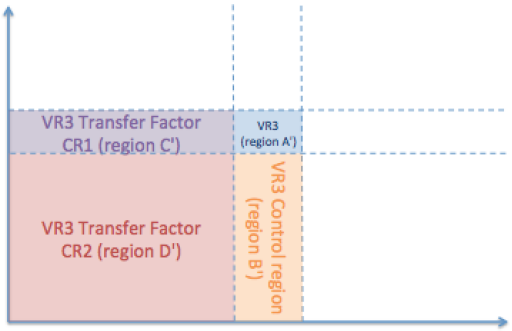
(a) Nominal regions



(b) Possible validation regions 1



(c) Possible validation regions 2



(d) Possible validation regions 3

Figure 10: Illustrations of the nominal signal and control regions, and possible validation and accompanying control regions. The ability to define the validation regions depends on the discreteness of the observables defining the plane.

5 From ABCD to the Matrix Method

5.1 The single-object case

In section 3 the ABCD method was expressed in terms of a free parameter $\tilde{m}$ that represented a transfer factor between region D and C (and, by assumption, region B and A). We will now see how the ABCD method can be seen as equivalent to the simplest case of the matrix method of background estimation.

In the simplest case of the matrix method, an event is flagged as *tight* (t) or *anti-tight* ($\bar{t}$) depending on whether a particular property of the event satisfies a requirement or not. The *tight* events are the signal region events and *anti-tight* events are the control region B events. The events in these two regions are either *real* (e.g. signal events) or *fake* (e.g. background events) in nature. The yield of tight ($N_t \equiv N_A$) and anti-tight ($N_{\bar{t}} \equiv N_B$) events are related to the underlying number of true but unknown real and fake events $\tilde{N}_R$ and $\tilde{N}_F$ by:

$$\begin{bmatrix} N_A \\ N_B \end{bmatrix} = \begin{bmatrix} r & f \\ 1-r & 1-f \end{bmatrix} \times \begin{bmatrix} \tilde{N}_R \\ \tilde{N}_F \end{bmatrix}, \tag{18}$$

where r and f are efficiencies for real and fake events to pass the tight event requirement. These efficiencies can be parameterized somehow, in which case the calculation can be repeated in the windows of the parameterization where the efficiencies are constant (i.e. within each bin of the parameterized efficiencies).

We will now see how this is identical to a certain case of the ABCD method. In the ABCD method with the presence of a signal and using the uniform model for the transfer factor, one has the following equations:

$$\begin{aligned} N_A &= \tilde{m}\tilde{N}_B^{\text{bkg}} + \tilde{\mu}N_A^{\text{sig}} \\ N_B &= \tilde{N}_B^{\text{bkg}} + \tilde{\mu}N_B^{\text{sig}} \\ N_C &= \tilde{m}\tilde{N}_D^{\text{bkg}} + \tilde{\mu}N_C^{\text{sig}} \\ N_D &= \tilde{N}_D^{\text{bkg}} + \tilde{\mu}N_D^{\text{sig}} \end{aligned} \tag{19}$$

We can define the following relationships:

$$\tilde{\mu} = \frac{\tilde{N}_R}{N_A^{\text{sig}} + N_B^{\text{sig}}} \tag{20}$$

$$r = \frac{N_A^{\text{sig}}}{N_A^{\text{sig}} + N_B^{\text{sig}}} \tag{21}$$

$$\tilde{N}_B^{\text{bkg}} = (1-\tilde{f})\tilde{N}_F \tag{22}$$

$$\tilde{m} = \frac{\tilde{f}}{1-\tilde{f}} \tag{23}$$

$$\tag{24}$$

and for simplicity we make an assumption that there is no signal contribution in region C or D (i.e. $N_C^{\text{sig}} = N_D^{\text{sig}} = 0$). We then find that the four equations defined by equation 19 become:

$$
\begin{aligned}
N_A &= \tilde{f}\tilde{N}_F + \tilde{r}\tilde{N}_R \\
N_B &= (1-\tilde{f})\tilde{N}_F + (1-\tilde{r})\tilde{N}_R \\
N_C &= \frac{\tilde{f}}{1-\tilde{f}}\tilde{N}_D^{\text{bkg}} \\
N_D &= \tilde{N}_D^{\text{bkg}}
\end{aligned} \tag{25}
$$

The first two equations are exactly the ones defined by the matrix method equation 18. The latter two equations effectively provide a measurement of the fake efficiency $\tilde{f}$: dividing one by the other leads to an estimate for $\tilde{f} = N_C/(N_C + N_D)$. We see that the ABCD method can be equivalent to the matrix method where the fake efficiency is determined by the events recorded in regions C and D, and the real efficiency is determined from the signal predictions in regions A and B.

5.2 Extending to multi-objects

Extending the matrix method to multiple objects amounts to further subdividing the data based on another discriminant. For example, N_A in equation 18 represented the events passing a tight selection. This could be divided into two sub-categories: N_{AA} and N_{AB}, where N_{AA} is the number of events that satisfy both tight selections, and N_{AB} is the number of events that satisfy the first tight selection but fail the second. A similar subcategorization can be applied to define N_{BA} and N_{BB} from the anti-tight events N_B.

$$
\begin{aligned}
N_{AB} &= \tilde{m}_1\tilde{N}_{BB}^{\text{bkg}} + \tilde{\mu}N_{AB}^{\text{sig}} \\
N_{BB} &= \tilde{N}_{BB}^{\text{bkg}} + \tilde{\mu}N_{BB}^{\text{sig}} \\
N_{AA} &= \tilde{m}_1\tilde{N}_{BA}^{\text{bkg}} + \tilde{\mu}N_{AA}^{\text{sig}} \\
N_{BA} &= \tilde{N}_{BA}^{\text{bkg}} + \tilde{\mu}N_{BA}^{\text{sig}}
\end{aligned}
$$

Appendix

In a paper, an appendix is used for technical details that would otherwise disturb the flow of the paper. Such an appendix should be printed before the Bibliography.

List of contributions

Auxiliary material

In an ATLAS paper, auxiliary plots and tables that are supposed to be made public should be collected in an appendix that has the title "Auxiliary material". This appendix should be printed after the Bibliography. At the end of the paper approval procedure, this information can be split into a separate document – see `atlas-auxmat.tex`.

In an ATLAS note, use the appendices to include all the technical details of your work that are relevant for the ATLAS Collaboration only (e.g. dataset details, software release used). This information should be printed after the Bibliography.